BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

KOREAN PATENT ABSTRACT (KR)

PUBLICATION

(51) IPC Code: H04L 12/26
(11) Publication No.: 2001-0064238 (43) Publication Date: July 9, 2001
(21) Application No.: 10-1999-0062388 (22) Application Date: December 27, 1999

(71) Applicant:
Electronics and Telecommunications Research Institute Oh, Kil-Rok
161, Gajung-dong, Yusung-gu, Daejeon, Korea

(72) Inventors:
LEE, HYUNG-SEOP; DO, HAN-CHUL; LEE, HYUNG-HO; LEE, SEUNG-SOO; SONG, SANG-SEOP

(54) Title of the Invention:

Apparatus and Method of Detecting Running Disparity Error

Abstract:

Provided are an apparatus and a method of detecting a running disparity (RD) detecting an error in a row of data received by a receiving unit in 8B/10B coding method to solve a problem that a sequential disparity generation method, which calculates disparity of 5B/6B and calculates the disparity of 8B/10B by adding the disparity of 5B/6B to 3B/4B coding disparity of 4 bits data, cannot be applied to a very high speed system such as gigabit Ethernet. The method detects a row of data that violates 8B/10B disparity from 10 bit data row using a clock of byte unit. That is, the number of 1 bits and 0 bits in 10 bit row is calculated, if the number of 1 bits is more than that of 0 bits, it means 'positive' status, if the number of 0 bits is more than that of 1 bits, it means 'negative' status, and if the number of 1 bits and the number of 0 bits are same, it means 'same' status. Previous RD value is calculated, and the previous RD value is compared to the number of '1' (RD6_ONE) in 6 bit row of present data row (RX_CDGR(n+1)) and the number of '1' (RD4_ONE) in 4 bit row. Therefore, the disparity violation can be detected directly in the 10 bit data row. Since the clock of byte unit is used, the method can be applied in the very high speed system such as the gigabit Ethernet.

(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) 。Int. Cl. [7]
H04L 12/26

(11) 공개번호 특2001-0064238
(43) 공개일자 2001년07월09일

(21) 출원번호 10-1999-0062388
(22) 출원일자 1999년12월27일

(71) 출원인 한국전자통신연구원
오길록
대전 유성구 가정동 161번지

(72) 발명자 이형섭
대전광역시중구문화2동극동아파트103-902
도한철
대전광역시유성구어은동한빛아파트109-1102
이형호
대전광역시유성구어은동한빛아파트108-1003호
이승수
전라북도전주시덕진구덕진동1가664-14전북대학교전자공학과부호연구실
송상섭
전라북도전주시덕진구덕진동1가664-14전북대학교전자공학과

(74) 대리인 전영일

심사청구 : 있음

(54) 러닝 디스패리티 에러 검출 장치 및 방법

요약

본 발명은 8B/10B 코딩 방법에 있어서 수신부에서 수신된 데이터열의 에러를 검출할 수 있는 러닝 디스패리티(RD; Running Disparity) 검출 장치 및 방법에 관한 것으로, 종래의 5B/6B 의 디스패리티를 구한 후 이 디스패리티 결과와 나머지 4 비트 데이터의 3B/4B 코딩의 디스패리티를 더하여 8B/10B 의 디스패리티를 구하는 순차적인 디스패리티 발생 방법으로는 기가비트 이더넷과 같은 초고속 시스템에 적용할 수 없었던 문제점을 해결하고자 하는 것이다. 본 발명은 바이트 단위 클럭을 이용하여 10 비트 데이터열에서 직접 8B/10B 의 디스패리티 위반 데이터열을 추출하는 방법을 제안한다. 즉, 본 발명은 10 비트열에서 '1'과 '0'의 비트수를 계산하여, '1'이 많은 경우 '포지티브', '0'이 많은 경우 '네가티브', '1'과 '0'의 비트수가 동일한 경우 '동일'로 나누어 처리하고, 위반 데이터열을 추출하는 RD 에러 검출 방법은 이전의 RD 값을 계산하고, 계산된 PRE_RD 값과 현재의 데이터열인 RX_CDGR(n+1)의 6 비트열의 '1'의 개수(RD6_ONE)와 4 비트열의 '1'의 개수(RD4_ONE)의 계산된 값을 비교하는 방법을 채택함으로써, 10 비트 데이터열에서 직접 디스패리티 위반을 추출할 수 있으며, 바이트 단위의 클럭 하나만을 사용하므로, 기가비트 이더넷과 같은 고속의

시스템에도 적용이 가능하다.

대표도
도 2

명세서

도면의 간단한 설명

도 1 은 종래의 RD 에러 검출 방법을 설명하는 도면.

도 2 는 본 발명에 의한 RD 에러 검출 장치를 나타내는 기능 블록도.

도 3 은 본 발명에서의 8B/10B 데이터열의 RD 에러를 검출하기 위하여 사용된 병렬 데이터열에 대하여 이전 데이터열인 RX_CDGR(n)과 현재의 데이터열인 RX_CDGR(n+1)의 입력 타이밍의 관계를 나타낸 도면.

도 4 는 도 2 의 PRE_RD 기능부(21)에서 RX_CDGR(n)의 병렬 데이터열을 이용하여 PRE_RD를 계산하는 과정을 나타낸 플로우챠트.

도 5 는 도 2 의 RD 에러 검사부(22)에서 RD 에러를 검출하는 과정을 플로우 챠트.

※ 도면의 주요부분에 대한 부호의 설명 ※

20: RD 에러 검출 장치

21: TOT_ONE 기능부

22: 바이트 플립플롭

23: RD_ONE 기능부

24: PRE_RD 기능부

25: RD 에러 검사부

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 8B/10B 의 러닝 디스패리티(Running Disparity; RD)의 RD 에러를 검출하는 방법에 관한 것으로, 더 상세히 말하면, 기가비트 이더넷 (Gigabit Ethernet) 시스템과 같이 초고속의 신호 처리가 요구되는 시스템에 적용되기에 적합한 RD 에러 검출 방법에 관한 것이다.

노드간의 신호 전송에 있어서 전송 효율을 높이기 위하여 NRZ(Non Return to Zero) 신호보다는 코딩된 신호를 사용한다. 데이터 통신에 가장 많이 사용되는 신호 코딩 방법으로는 맨체스터(manchester), 4B/5B, 3B/4B, 또는 8B/10B 등이 있다. 8B/10B 코딩의 RD 에러 검출은 물리 계층인 PCS 수신 기능부의 중요한 기술중의 하나로서, 종래에는, 8

B/10B 코딩에 있어서, 코드의 에러를 파악하기 위한 러닝 디스패리티 검출 방법으로 종래에 사용되는 방법은 먼저 5B/6B 의 디스패리티를 구한 후, 이 디스패리티 결과를 반영하여 3B/4B 의 디스패리티를 발생시키고 이 디스패리티를 검사하여 8B/10B 의 디스패리티를 구하는 방식을 사용하였다. 이 경우, 6비트열의 데이터를 이용하여 발생시킨 5B/6B 의 디스패리티 결과는 다시 3B/4B 코딩에 의하여 발생된 4 비트열에 반영하도록 되어 있다.

이와 같은 순차적인 디스패리티 발생 방법에 의하여 8B/10B 코드의 에러를 검출하는 방법은 각 5B/6B 및 3B/4B 디스패리티 발생부에서 수신 데이터의 전송 속도보다 2배 빠른 비트 단위의 클럭이 요구되고, 최종 8B/10B 의 디스패리티를 검사하기 위하여 수신 데이터의 전송 속도와 동일한 바이트 클럭이 요구된다. 따라서 순차적인 디스패리티 발생 방법에 의한 RD 에러 검출 방법은 수신 데이터 전송 속도보다 빠른 클럭이 요구되므로, 구현이 복잡하다.

또한, 기가비트 이더넷 시스템과 같이 초고속으로 처리되어야 하는 시스템에 있어서는 RD 에러 검출 시에 클럭의 타이밍을 맞추기 어려우므로, RD 에러 검출 시 에러 발생 확률이 높으며, RD를 계산하기 위하여 비트 단위 및 바이트 단위의 2가지 클럭이 요구된다.

이하, 도 1을 참조하여 종래의 RD 에러 검출 방법을 설명한다.

도 1은 종래의 8B/10B 코드에서 8B/10B 데이터열의 RD 에러 검출 방법을 나타내는 도면이다. 도 1 에 도시한 종래의 RD 에러 검출 방법은 먼저 5B/6B 의 디스패리티를 구한 후, 이 디스패리티 결과를 반영하여 3B/4B 디스패리티를 발생시키고, 이 디스패리티를 검사하여 8B/10B 의 디스패리티를 구하는 방법이다.

도 1 에 도시된 RD 에러 검출 방법의 동작을 살펴보면, 수신되는 데이터열의 순서대로 RD를 계산하기 위하여, 먼저 수신되는 6 비트열의 RD를 5B/6B 디스패리티 기능부(11)에서 계산하고, 계산된 신호(PDSP)를 3B/4B 디스패리티 기능부(12)로 입력한다.

3B/4B 디스패리티 기능부(12)는 5B/6B 디스패리티 기능부(11)로부터 입력된 신호(PDSP)와 나머지 잔존하는 4 비트열과 조합하여 디스패리티를 계산하고 8B/10B 코드의 최종 RD 값인 신호(NDSP)를 발생시킨다. 이렇게 발생된 디스패리티의 신호(NDSP)값은 8B/10B 코드의 데이터열에 에러가 발생했는지를 파악하는 신호로서, RD 에러 검사부(13)에 입력된다. RD 에러 검사부(13)는 신호(NDSP)를 입력받아 데이터열에 에러가 발생했는지를 결정한다.

도 1 의 종래 방법에서는, 5B/6B 디스패리티 기능부(11)와 3B/4B 디스패리티 기능부(12)에는 수신 데이터(10 비트열)의 전송 속도보다 최소 2배 빠른 비트 단위 클럭(BITCLK)이 공급되고 있으며, RD 에러 검사부(13)에는 수신 데이터(10 비트열)의 전송 속도와 동일한 바이트 단위 클럭(BYTECLK)이 공급되고 있다. 이와 같이, 바이트 클럭과 비트 클럭이 별도로 요구되는 이유는 5B/6B 의 디스패리티를 구한 후, 이 디스패리티 결과를 반영하여 3B/4B 디스패리티를 발생시키는 순차적인 방법을 사용하기 때문이다.

그런데, 수신 데이터의 한 주기에서 디스패리티를 파악하기 위하여 비트 단위 클럭(BITCLK)은 바이트 단위 클럭(BYTECLK)에 대하여 최소 2배 빠른 클럭이어야 한다. 따라서, 도 1 의 RD 검출 방법은 순차적인 방법에 의하여 RD 에러를 검사하기 때문에 수신 데이터의 전송 속도보다 최소 2배 빠른 클럭이 요구되고, 초고속으로 처리하는 경우 디스패리티 발생에 있어서, 타이밍 위반의 문제가 발생할 확률이 높아지는 문제점이 있다.

발명이 이루고자 하는 기술적 과제

따라서, 본 발명은 상기와 같은 종래기술의 문제점을 해결하기 위하여 안출된 것으로서, 기가비트 이더넷과 같이 초고속 시스템에 적용할 수 있는 RD 에러 검출 방법을 제공하는 것을 목적으로 한다.

발명의 구성 및 작용

상기한 목적을 달성하기 위하여, 본 발명의 한 형태에 의하면, 이전의 데이터열을 입력받아 그 데이터열의 '1'의 개수를 계산하여 TOT_ONE 값을 생성하는 TOT_ONE 기능부와, 이전의 데이터열을 입력받아 현재의 데이터열을 생성하는 바이트 플립플롭과, 바이트 플립플롭에 접속되어 3B/4B 비트열과 5B/6B 비트열의 '1'의 개수를 계산하여 RD4_ONE 값과 RD6_ONE 값을 생성하는 RD_ONE 기능부와, TOT_ONE 기능부 및 RD_ONE 기능부에 접속되어, TOT_ONE 값과 RD4_ONE 값을 이용하여 이전의 RD 값인 PRE_RD 값을 생성하는 PRE_RD 기능부와, PRE_RD 기능부, RD_ONE 기능부에 접속되어, PRE_RD 값, RD4_ONE 값, 및 RD6_ONE 값을 이용하여, 데이터열에 RD 에러가 있는지를 검출하는 RD 에러 검사부를 구비하는 RD 에러 검출 장치를 제공한다.

본 발명의 다른 형태에 의하면, 이전의 데이터열의 '1'의 개수를 계산하여 TOT_ONE 값을 생성하는 제 1 단계와, 현재의 데이터열을 사용하여 3B/4B 비트열과 5B/6B 비트열의 '1'의 개수를 각각 계산하여 RD4_ONE 값과 RD6_ONE 값을 생성하는 제 2 단계와, TOT_ONE 값과 RD4_ONE 값을 이용하여 이전의 RD 값인 PRE_RD 값을 생성하는 제 3 단계와, PRE_RD 값, RD4_ONE 값, 및 RD6_ONE 값을 이용하여, 데이터열에 RD 에러가 있는지를 검출하는 제 4 단계를 구비하는 RD 에러 검사 방법을 제공한다.

바람직한 실시예에 있어서, 제 3 단계는, (3-A) TOT_ONE 값이 5보다 큰지를 판정하여 PRE_RD 값을 생성하는 단계와, (3-B) RD_ONE 값이 2보다 큰지를 판정하여 PRE_RD 값을 생성하는 단계를 구비하는 것이 바람직하다.

또한, (3-A) 단계에서, TOT_ONE 값이 5보다 크면, PRE_RD 값으로서 '1'을 출력하고, TOT_ONE 값이 5보다 작으면, PRE_RD 값으로서 '0'을 출력하고, TOT_ONE 값이 5이면, (3-B) 단계로 진행하고, (3-B) 단계에서, RD4_ONE 값이 2보다 크면, PRE_RD 값으로서, '1'을 출력하고, RD4_ONE 값이 2보다 작으면, PRE_RD 값으로서, '0'을 출력하고, RD4_ONE 값이 2이면, PRE_RD 값으로서, 이전의 PRE_RD 값인 'PRE_RD(n-1)'을 출력하는 것이 바람직하다.

또한, 제 4 단계는, (4-A) 제 3 단계에서 생성된 PRE_RD 값을 판정하는 단계와, (4-B) 현재의 데이터열인 RX_CDGR(n+1)의 6 비트열의 '1'의 개수인 RD6_ONE 값을 판정하는 단계와, (4-C) 현재의 데이터열인 RX_CDGR(n+1)의 4 비트열의 '1'의 개수인 RD4_ONE 값을 판정하는 단계를 구비하는 것을 특징으로 하는 단계를 구비하는 것이 바람직하다.

이하, 첨부된 도면을 참조하여 본 발명의 실시예를 설명하면 다음과 같다.

도 2 는 본 발명에 따른 10 비트 병렬 데이터열에 의한 RD 에러 검출 장치의 기능 블록도이다. RD 에러 검출 장치(20)는 TOT_ONE 기능부(21), 바이트 플립플롭(22), RD_ONE 기능부(23), PRE_RD 기능부(24), 및 RD 에러 검사부(25)로 구성되어 있다.

TOT_ONE 기능부(21)는 RX_CDGR(n)[9:0]의 데이터열을 입력받아 그 데이터열의 전체 '1'의 개수를 계산하여 TOT_ONE 값을 생성한다. TOT_ONE 기능부(21)에서 생성된 TOT_ONE 값은 PRE_RD 기능부(24)로 전달된다.

또한, 이전 데이터열과 현재의 데이터열을 형성하기 위하여 RX_CDGR(n)[9:0]의 데이터열을 바이트 플립플롭(22)을 통과시켜 RX_CDGR(n+1)[9:0]의 10 비트 데이터열을 형성하여 RD_ONE 기능부(23)로 전달하고, 10 비트 데이터열중 6 비트열 RX_CDGR(n+1)[5:0]는 RD 에러 검사부(25)에 전달한다.

그리고, RD_ONE 기능부(23)는 RD_CDGR(n+1)[9:0]의 데이터열을 사용하여 3B/4B 비트열과 5B/6B 비트열의 '1'의 개수를 계산하여 RD4_ONE 값과 RD6_ONE 값을 생성한다. RD_ONE 기능부(23)에서 발생된 RD4_ONE 값은 PRE_RD 기능부(24)와 RD 에러 검사부(25)에 입력되고, RD6_ONE 값은 RD 에러 검사부(25)에 입력된다.

PRE_RD 기능부(24)는 TOT_ONE 값과 RD4_ONE 값을 사용하여 후술할 도 4와 같은 PRE_RD 계산과정을 수행하여 PRE_RD 값을 계산하여 이를 RD 에러 검사부(25)에 전달한다.

그후, RD 에러 검사부(25)는 RX_CDGR(n+1)[5:0]의 데이터열과 PRE_RD 값과 RD4_ONE 값과 RD6_ONE 값을 사용하여 후술한 도 5 의 RD 에러 계산 과정을 수행하여 RDERR 값을 출력한다. 여기서, RD 위반인 경우에는 RDERR 값을 '1'로 출력한다. RD 에러 검사부(25)에서 RD를 계산하는 방법으로 '1'의 비트수와 '0'의 비트수를 비교하여 판단한다.

여기서, RX-CDGR(n)[9:0] 데이터열의 비트 순서는 [j h g f i e d c b a]를 가지며, RD6_ONE을 계산하기 위해서 RX_CDGR(n+1)[5:0] ="[i" e d c b a] 6 비트열을 이용하여 계산하고, RD4_ONE을 계산하기 위해 RX_CDGR(n+1)[9:0] ="[j" h g f] 4 비트열을 이용하여 각각 계산한다.

도 3 은 본 발명에서의 8B/10B 데이터열의 RD 에러를 검출하기 위하여 사용된 병렬 데이터열에 대하여 이전 데이터열인 RX_CDGR(n)과 현재의 데이터열인 RX_CDGR(n+1)의 입력 타이밍의 관계를 나타낸 것이다. 예로서, 기가비트 이더넷 시스템에서 수신되는 병렬 데이터열의 간격은 8ns 이다.

도 4 는 도 2 의 PRE_RD 기능부(21)에서 RX_CDGR(n)의 병렬 데이터열을 이용하여 PRE_RD를 계산하는 과정을 플로우챠트로 나타낸 것이다. PRE_RD 신호는 이전 코드 그룹의 RD 값을 의미하며, PRE_RD 기능부(21)는 수신 데이터로서 RX_CDGR(n)[9:0]을 사용하고 이전 코드 그룹의 '1' 비트 개수의 정보를 가지고 있는 내부 신호 TOT_ONE와 RD4_ONE를 이용하여 계산을 수행한다. 도 4에서, 포지티브인 경우('1'의 비트수> '0'의 비트수)는 전체 검사 비트열에서 '1'의 비트수가 '0'의 비트수보다 많은 경우를 의미한다. 네가티브인 경우('1'의 비트수< '0'의 비트수)는 '1'의 비트수가 '0'의 비트수보다 적은 경우를 의미한다. 동일한 경우('1'의 비트수="'0'의" 비트수)는 '1'의 비트수가 '0'의 비트수와 동일한 경우를 표시한다.

이하, 도 4를 참조하여 PRE_RD를 계산하기 위한 수행 과정을 설명한다.

먼저, 단계(41)에서, 10 비트의 RX_CDGR(n)[9:0]에서 '1'의 개수를 세어 TOT_ONE값을 구한다. 만약, TOT_ONE 값이 5보다 작으면 (즉, '1'의 비트수가 '0'의 비트수보다 적음), 네가티브이므로, PRE_RD 값으로서 '0'을 출력하고, 반대로 TOT_ONE 값이 5보다 크면('1'의 비트수가 '0'이 비트수보다 많음), 포지티브이므로, PRE_RD 값으로서 '1'을 출력한다.

TOT_ONE 값이 5인 경우(즉, '1'의 비트수가 '0'의 비트수와 동일한 경우)에는 '1'과 '0'이 균등한 데이터열이기 때문에 즉시 PRD_RD를 계산할 수 없고, 이러한 경우에는 3B/4B 의 RD4_ONE 값에 따라 PRE_RD 값이 결정되어야 하므로, 단계(42)로 진행한다.

단계(42)에서, RD4_ONE 값이 네가티브인 경우(RD4_ONE 값이 2보다 적은 경우)에는 PRE_RD 값으로서 '0'을 출력하고, RD4_ONE 값이 포지티브인 경우(RD4_ONE 값이 2보다 많은 경우)에는 PRE_RD 값으로서 '1'을 출력한다. 또한, RD4_ONE 값이 동일한 경우('1'과 '0'의 비트수가 동일한 경우)에는 PRE_RD 값은 이전의 PRE_RD 값(PRE_RD(n-1)값)과 같게 된다. 또한, PRE_RD값을 계산하는 과정에서 5B/6B 인 RD6_ONE 값을 고려하지 않아도 되는 이유는 최종 RD 값은 3B/4B 에 의해 정해지기 때문이다.

도 5 는 도 2 의 RD 에러 검사부(22)에서 RD 에러를 검출하는 과정을 플로우 챠트로 표시한 것이다. 도 5 의 플로우챠트에서는 각각의 조건문(51-59)에 따라 경로가 결정되며, 최종적인 RD 에러 결과는 'E'(Error="'1')" 또는 'N'(No error="'0')이" 된다. 최종 판정된 결과값이 'E' 이면, RD 에러를 의미하고, 'N'이면, RD 에러가 아님을 의미한다. RD 에러 판단은 이전의 RD 값인 PRE_RD 값에 따라 조건이 달라지며, 기본 성질은 이전의 RD 값이 '포지티브'이면, 현재의 코드열은 '네가티브'이거나 '동일'이어야 하며, 반대로 이전의 RD 값이 '네가티브'이면, 현재의 코드열은 '포지티브'이거나 '동일'이어야 하며, 이때 올바른 RD 값을 갖는 코드열이 수신되고 있다고 판단한다. 여기서, '포지티브'와 '네가티브'와 '동일'의 정의는 도 4에서 정의한 것과 동일하다.

이하, 도 5를 참조하여 RD 에러검사부(22)에서의 RD 에러 검출 과정에 대하여 설명한다.

먼저, 단계(51)에서, 도 4에서 구한 PRE_RD 값이 '0'인지를 판정한다. PRE_RD 값이 '0'이면, 단계(52)로 진행하여, RX_CDGR(n+1)[5:0] 의 '1'의 개수를 의미하는 RD6_ONE 값을 구한다. 단계(52)에서, RD6_ONE 값이 '네가티브'이면(즉, RD6_ONE 값이 3보다 작으면), PRE_RD 값도 네가티브이고 바로 다음에 수신되는 데이터열의 RD 도 네가티브이므로, RD 에러로 판단하여 'E'를 출력한다.

반대로, RD6_ONE 값이 '포지티브'이면(즉, RD6_ONE 값이 3보다 크면), 현재까지는 RD 에러가 아니므로, 단계(57)로 진행하여, 나머지 RX_CDGR(n+1)[9:6] 의 '1'의 개수를 나타내는 RD4_ONE 값으로 판단해야 한다. 단계(57)에서, RD4_ONE 값이 '포지티브'이면, RD 에러로 판정하여 'E'를 출력하고, RD4_ONE 값이 '네가티브'이거나 '동일'이면, 올바른 데이터열이 수신된 것으로 판정하여 'N'을 출력한다.

단계(52)에서, RD6_ONE 값이 '동일'이면(즉, RD6_ONE 값이 3이면), 단계(54)로 진행하여 RX_CDGR(n+1)[5:0] 이 " 111000" 인지를 판정한다. 단계(54)에서, RX_CDGR(n+1) [5:0] 이 " 111000" 이면, RD 에러로 판정하여 'E'를 출력한다. 그렇지 않으면, 현재까지는 RD 에러가 아니므로, 단계(56)으로 진행한다.

단계(56)에서, 3B/4B 인 RX_CDGR(n+1)[9:6] 데이터열의 '1'의 개수인 RD4_ONE 값을 판단한다. 단계(56)에서, RD4_ONE 값이 '네가티브'이면(즉, RD4_ONE 값이 2보다 작으면), PRE_RD 도 '네가티브'이고 현재의 RD 값도 '네가티브'가 되므로, RD 에러로 판정하여, 'E'를 출력한다. RD4_ONE 값이 '포지티브'이거나 '동일'이면(즉, RD4_ONE 값이 2보다 크거나 동일하면), 올바른 데이터열이 수신된 것으로 판정하여, 'N'을 출력한다.

단계(51)에서, PRE_RD 값이 '1'이면, 단계(53)으로 진행하여 RD6_ONE 값을 판정한다. 단계(53)에서, RD6_ONE 값이 '포지티브'이면(즉, RD6_ONE 값이 3보다 크면), 이전의 RD 값이 '포지티브'이고 현재의 RD 값도 '포지티브'이므로, RD 에러로 판정하여, 'E'를 출력한다. 반대로, RD6_ONE 값이 '네가티브'이면(즉, RD6_ONE 값이 3보다 작으면), 현재까지는 RD 에러가 아니므로, 단계(58)로 진행하여 3B/4B 데이터열의 '1'의 개수를 나타내는 RD4_ONE 값을 판정해야 한다.

단계(58)에서, RD4_ONE 값이 '네가티브'이면(즉, RD4_ONE 값이 2보다 작으면), RD 에러로 판정하여 'E'를 출력하고, 그렇지 않으면, 즉, RD4_ONE 값이 '포지티브'이거나 '동일'이면(RD4_ONE 값이 2보다 크거나 동일하면), 올바른 데이터열이 수신된 것으로 판정하여, 'N'을 출력한다.

또한, 단계(53)에서, RD6_ONE 값이 '동일'이면(즉, RD6_ONE 값이 3이면), 단계(55)로 진행한다. 단계(55)에서, RX_CDGR(n+1)[5:0] 데이터열이 " 000111" 인지를 판정하여, RX_CDGR(n+1) [5:0] 데이터열이 " 000111" 이면, RD 에러로 판정하여 'E'를 출력하고, 그렇지 않으면, 단계(59)로 진행하여 RD4_ONE 값을 판정한다.

단계(59)에서, RD4_ONE 값이 '포지티브'이면(즉, RD4_ONE 값이 2보다 크면), 이전의 RD 값도 '포지티브'이고 현재의 RD 값도 '포지티브'이므로, RD 에러로 판정하여 'E'를 출력하고, RD4_ONE 값이 '동일'이거나 '네가티브'이면(즉, RD4_ONE 값이 2이거나 작으면), RD 에러가 발생하지 않은 것으로 판정하여 'N'을 출력한다.

위에서 양호한 실시예에 근거하여 이 발명을 설명하였지만, 이러한 실시예는 이 발명을 제한하려는 것이 아니라 예시하려는 것이다. 이 발명이 속하는 분야의 숙련자에게는 이 발명의 기술사상을 벗어남이 없이 위 실시예에 대한 다양한 변화나 변경 또는 조절이 가능함이 자명할 것이다. 그러므로, 이 발명의 보호범위는 첨부된 청구범위에 의해서만 한정될 것이며, 위와 같은 변화예나 변경예 또는 조절예를 모두 포함하는 것으로 해석되어야 할 것이다.

발명의 효과

이상과 같이 본 발명에 의하면, 종래의 방법에서 5B/6B 디스패리티를 구하고 3B/4B 디스패리티를 순차적으로 처리하기 위하여 비트 단위 및 바이트 단위의 클럭이 필요하였던 것에 비하여, 본 발명은 8B/10B 병렬 데이터열에서 직접적으로 디스패리티를 계산하고, '1'의 개수를 계산하는 기능에서 병렬 데이터열의 바이트 클럭 하나만을 사용함으로써, 종래의 방법에서 '1'의 개수를 계산시 비트 단위 클럭을 사용함으로써 발생하는 계산 타이밍의 위반을 피하고 초고속으로 RD 에러를 검출할 수 있는 효과가 있다.

(57) 청구의 범위

청구항 1.

이전의 데이터열을 입력받아 그 데이터열의 '1'의 개수를 계산하여 TOT_ONE 값을 생성하는 TOT_ONE 기능부와,

이전의 데이터열을 입력받아 현재의 데이터열을 생성하는 바이트 플립플롭과,

상기 바이트 플립플롭에 접속되어 3B/4B 비트열과 5B/6B 비트열의 '1'의 개수를 계산하여 RD4_ONE 값과 RD6_ONE 값을 생성하는 RD_ONE 기능부와,

상기 TOT_ONE 기능부 및 RD_ONE 기능부에 접속되어, 상기 TOT_ONE 값과 RD4_ONE 값을 이용하여 이전의 RD 값인 PRE_RD 값을 생성하는 PRE_RD 기능부와,

상기 PRE_RD 기능부 및 RD_ONE 기능부에 접속되어, 상기 PRE_RD 값, RD4_ONE 값 및 RD6_ONE 값을 이용하여 데이터열에 RD 에러가 있는지를 검출하는 RD 에러 검사부를 구비하는 것을 특징으로 하는 RD 에러 검출 장치.

청구항 2.

이전의 데이터열의 '1'의 개수를 계산하여 TOT_ONE 값을 생성하는 제 1 단계와,

현재의 데이터열을 사용하여 3B/4B 비트열과 5B/6B 비트열의 '1'의 개수를 각각 계산하여 RD4_ONE 값과 RD6_ONE 값을 생성하는 제 2 단계와,

상기 TOT_ONE 값과 상기 RD4_ONE 값을 이용하여 이전의 RD 값인 PRE_RD 값을 생성하는 제 3 단계와,

상기 PRE_RD 값, RD4_ONE 값 및 RD6_ONE 값을 이용하여, 데이터열에 RD 에러가 있는지를 검출하는 제 4 단계를 구비하는 것을 특징으로 하는 RD 에러 검사 방법.

청구항 3.

제 2 항에 있어서, 상기 제 3 단계는,

(3-A) 상기 TOT_ONE 값이 5보다 큰지를 판정하여 PRE_RD 값을 생성하는 단계와,

(3-B) 상기 RD_ONE 값이 2보다 큰지를 판정하여 PRE_RD 값을 생성하는 단계를 구비하는 것을 특징으로 하는 RD 에러 검사 방법.

청구항 4.

제 3 항에 있어서, 상기 (3-A) 단계에서,

상기 TOT_ONE 값이 5보다 크면, PRE_RD 값으로서 '1'을 출력하고,

상기 TOT_ONE 값이 5보다 작으면, PRE_RD 값으로서 '0'을 출력하고,

상기 TOT_ONE 값이 5이면, 상기 (3- B) 단계로 진행하고,

상기 (3-B) 단계에서,

상기 RD4_ONE 값이 2보다 크면, PRE_RD 값으로서, '1'을 출력하고,

상기 RD4_ONE 값이 2보다 작으면, PRE_RD 값으로서, '0'을 출력하고,

상기 RD4_ONE 값이 2이면, PRE_RD 값으로서 이전의 PRE_RD 값인 'PRE_RD(n-1)'을 출력하는 것을 특징으로 하는 RD 에러 검출 방법.

청구항 5.

제 2 항에 있어서, 상기 제 4 단계는,

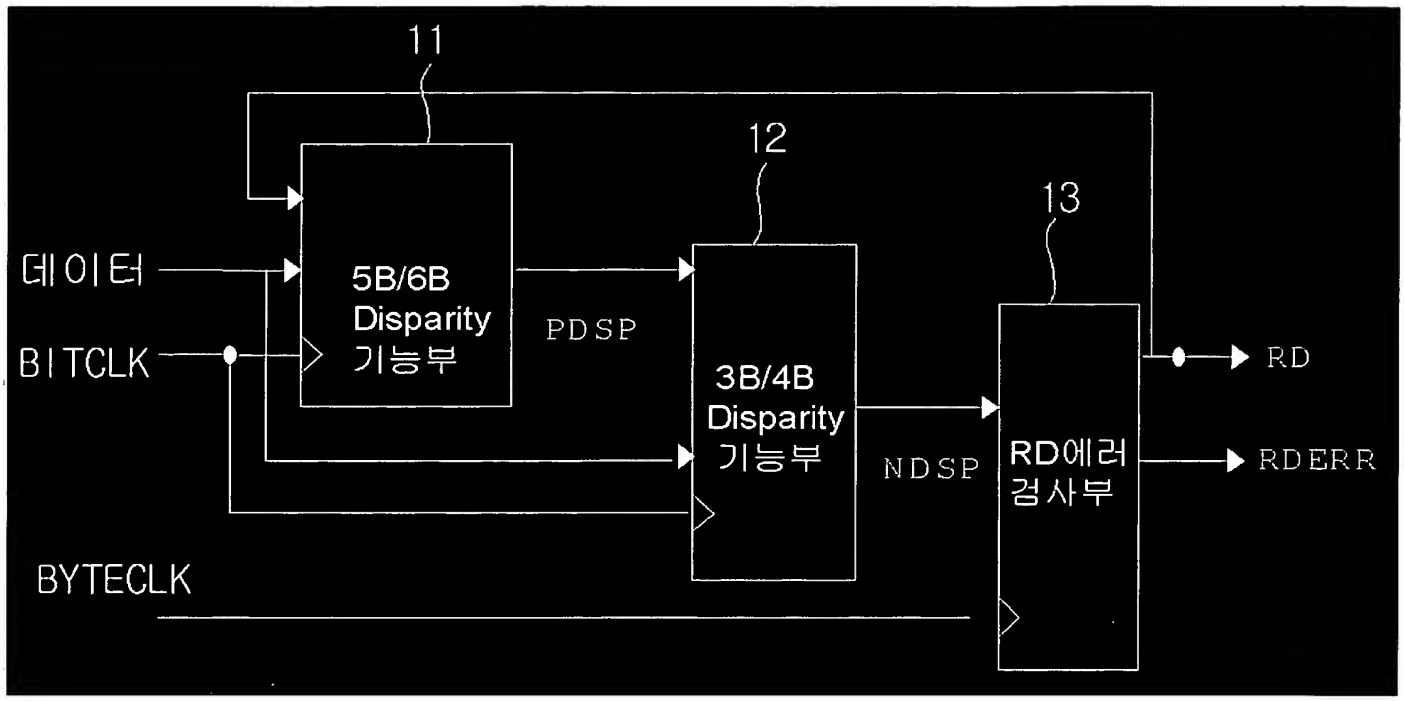
(4-A) 상기 제 3 단계에서 생성된 상기 PRE_RD 값을 판정하는 단계와,

(4-B) 현재의 데이터열인 RX_CDGR(n+1)의 6 비트열의 '1'의 개수인 RD6_ONE 값을 판정하는 단계와,

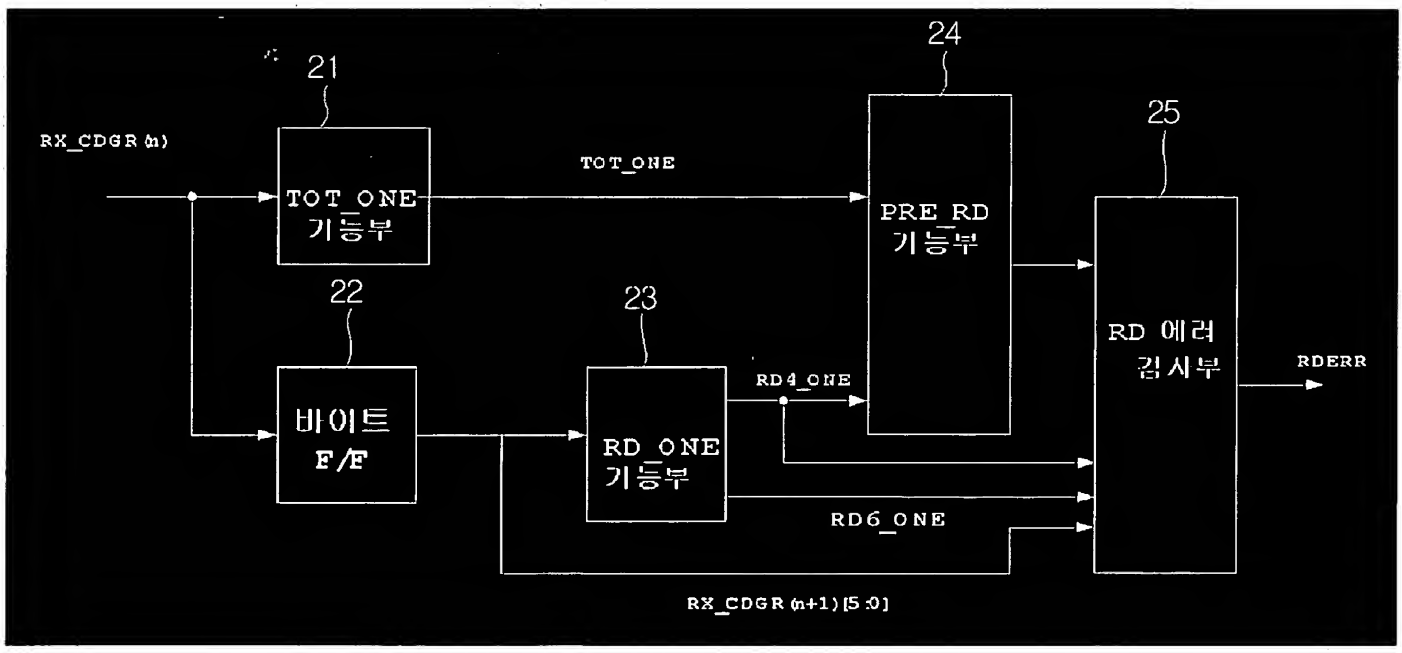
(4-C) 현재의 데이터열인 RX_CDGR(n+1)의 4 비트열의 '1'의 개수인 RD4_ONE 값을 판정하는 단계를 구비하는 것을 특징으로 하는 단계를 구비하는 것을 특징으로 하는 RD 에러 검출 방법.

도면

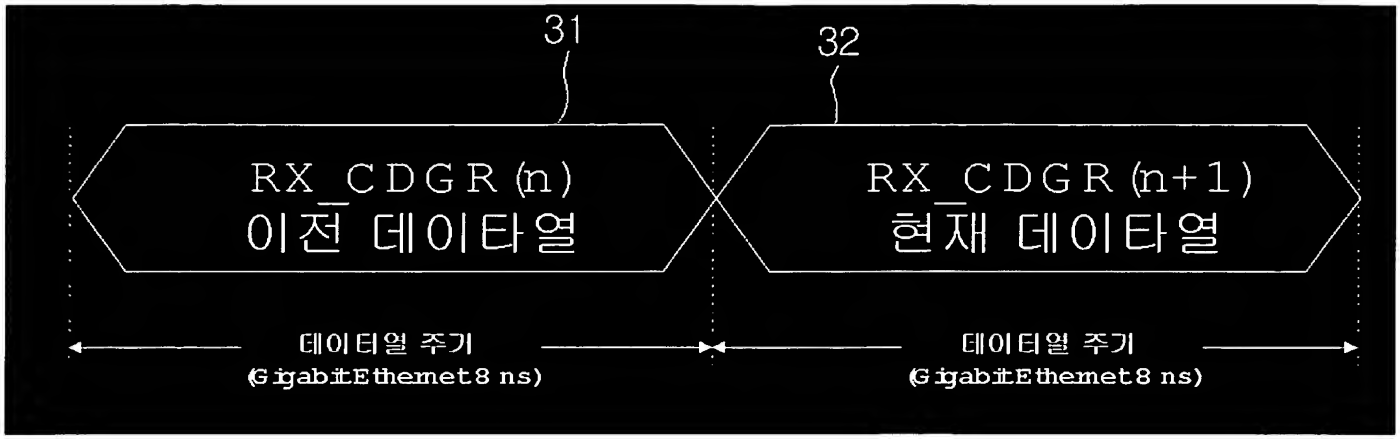
도면 1


11
12
13
데이터
BITCLK
5B/6B
Disparity
기능부
PDSP
3B/4B
Disparity
기능부
NDSP
RD에러
검사부
RD
RDERR
BYTECLK

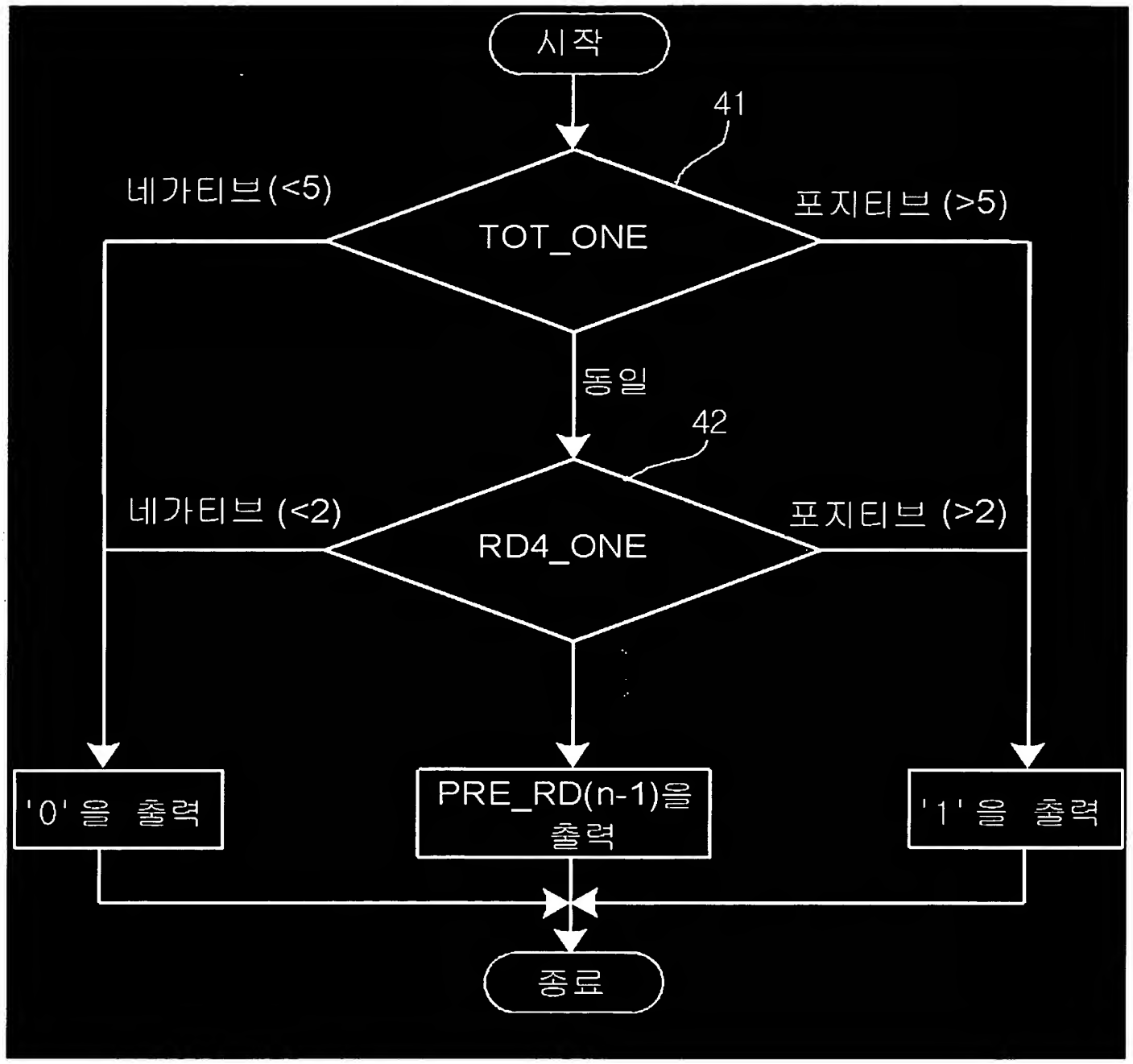
도면 2


21
24
25
RX_CDGR(n)
TOT_ONE
기능부
TOT_ONE
PRE_RD
기능부
22
23
RD_ONE
기능부
RD4_ONE
RD 에러
검사부
RDERR
바이트
F/F
RD6_ONE
RX_CDGR(n+1)[5:0]

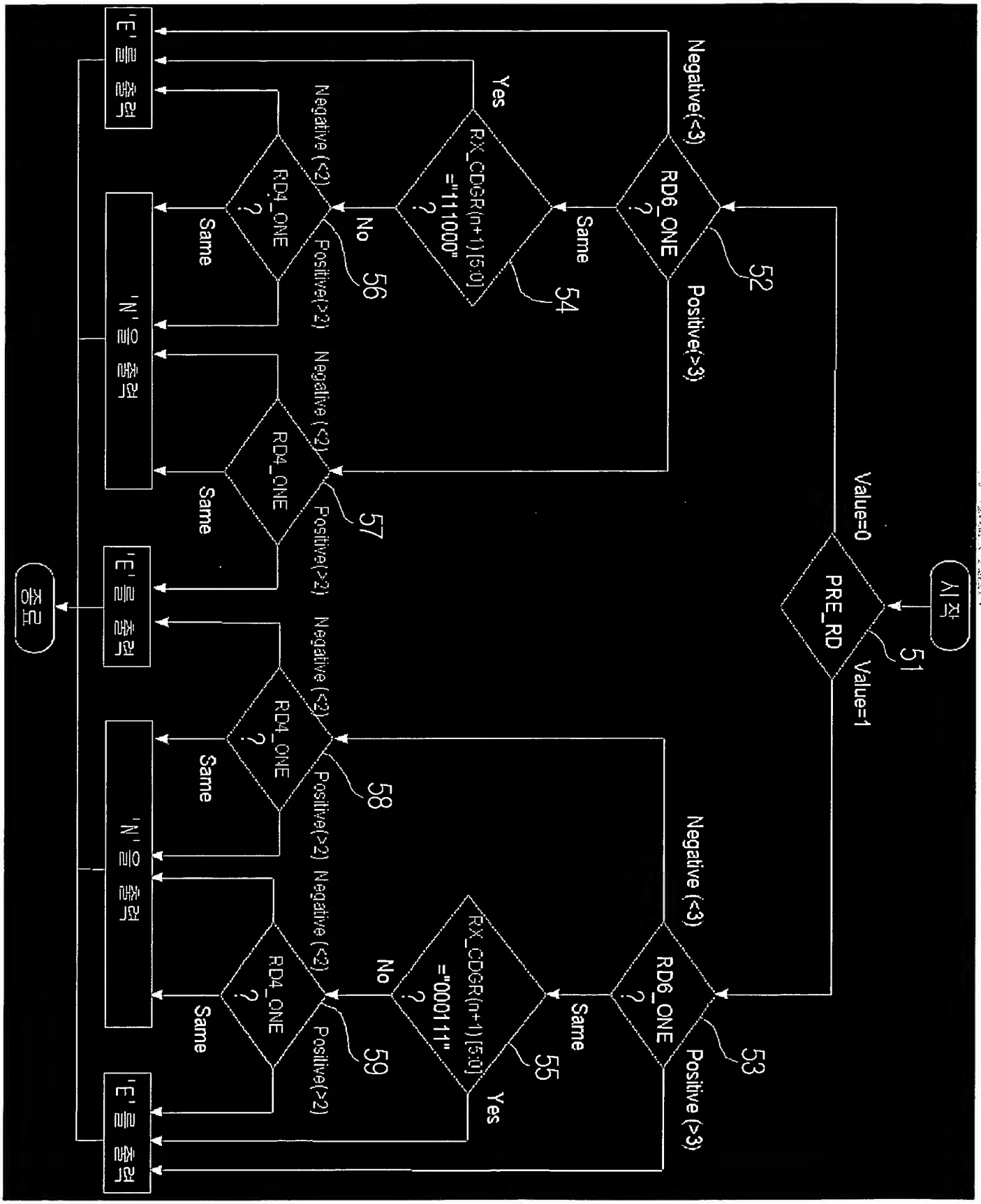
도면 3


31
32
RX_CDGR(n)
이전 데이타열
RX_CDGR(n+1)
현재 데이타열
데이터열 주기
(GigabitEthernet8 ns)
데이터열 주기
(GigabitEthernet8 ns)

도면 4


시작
41
네가티브(<5)
TOT_ONE
포지티브 (>5)
동일
42
네가티브 (<2)
RD4_ONE
포지티브 (>2)
'0'을 출력
PRE_RD(n-1)을 출력
'1'을 출력
종료

도면 5


시작
51
PRE_RD
Value=0
Value=1
RD6_ONE
?
52
Negative(<3)
Positive(>3)
RX_CDGR(n+1)[5:0]
="111000"
?
54
Yes
Same
No
56
RD4_ONE
?
Negative (<2)
Same
Positive(>2)
'N'을 출력
'E'를 출력
57
RD4_ONE
?
Negative (<2)
Same
Positive(>2)
'E'를 출력
종료
58
RD4_ONE
?
Negative (<2)
Same
Positive(>2)
'N'을 출력
RD6_ONE
?
53
Negative (<3)
Positive (>3)
RX_CDGR(n+1)[5:0]
="000111"
?
55
Yes
Same
No
59
RD4_ONE
?
Negative (<2)
Same
Positive(>2)
'E'를 출력